

**Факультет прикладних інформаційних технологій  
та електроінженерії**

Кафедра автоматизації технологічних  
процесів і виробництв

**МЕТОДИЧНІ ВКАЗІВКИ**  
до лабораторної роботи №23  
«Розробка та відлагодження програми керування  
технологічним обладнанням у середовищі програмного  
забезпечення «Factory I/O»-CODESYS»

з курсу «Проектування систем автоматизації»

для студентів спеціальності 151 – «Автоматизація та  
комп'ютерно-інтегровані технології»

Тернопіль 2018

**Міністерство освіти і науки України  
Тернопільський національний технічний університет  
ім. Івана Пулюя**

**Факультет прикладних інформаційних технологій  
та електроінженерії  
Кафедра автоматизації технологічних  
процесів і виробництв**

**МЕТОДИЧНІ ВКАЗІВКИ  
до лабораторної роботи №23  
«Розробка та відлагодження програми керування технологічним  
обладнанням у середовищі програмного забезпечення  
«Factory I/O»-CODESYS»**

**з курсу «Проектування систем автоматизації»**

**для студентів спеціальності 151 – «Автоматизація та комп'ютерно-  
інтегровані технології»**

**Тернопіль, 2018**

Методичні вказівки до лабораторної роботи № 23. Розробка та відлагодження програми керування технологічним обладнанням у середовищі програмного забезпечення Factory I/O - CODESYS з курсу «Проектування систем автоматизації» / Шкодзінський О.К., Пісьціо В.П., Сікора Д.А. - Тернопіль: ТНТУ, 2018 - 18 с.

Для студентів спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології.

Укладачі: Шкодзінський О.К., Пісьціо В.П., Сікора Д.А.

Методичні вказівки розглянуті, схвалені і затверджені на засіданні кафедри автоматизації технологічних процесів та виробництв Тернопільського національного технічного університету імені Івана Пулюя (протокол № 11 від 26 лютого 2018 року).

**Тема:** Розробка та відлагодження програми керування технологічним обладнанням у середовищі програмного забезпечення «Factory I/O»-CODESYS.  
**Мета:** Навчитися підключати готову сцену з технологічним обладнанням у «Factory I/O» до віртуального програмованого логічного контролера та отримати навички його програмування з використанням мови структурованого тексту (ST) у середовищі CODESYS.

## 1. ТЕОРЕТИЧНІ ВІДОМОСТІ

У цій лабораторній роботі показано, як використовувати CODESYS разом із «Factory I/O» для моделювання роботи автоматизованого обладнання на базі OPC Data Access. Дотримуючись інструкцій нижче, слід створити новий проект автоматизованого обладнання з відлагодженням у середовищі CODESYS програми керування для програмованого логічного контролера. Як приклад буде розглядатись сцена - «Sorting by Height».

## 2. ХІД РОБОТИ

### Перше заняття

2.1. Запустити ПЗ «Factory I/O».

2.2. Відкрити готову сцену з технологічним обладнанням з бібліотеки. На панелі інструментів слід з меню **File** вибрати **Open**. У меню вибору сцен натиснути ЛКМ на **Scenes**. Сцену обрати згідно з варіантом (див. розділ 3, даних методичних вказівок) або скористатись прикладом. У випадку, що розглядається, прикладом є «Sorting by Height».

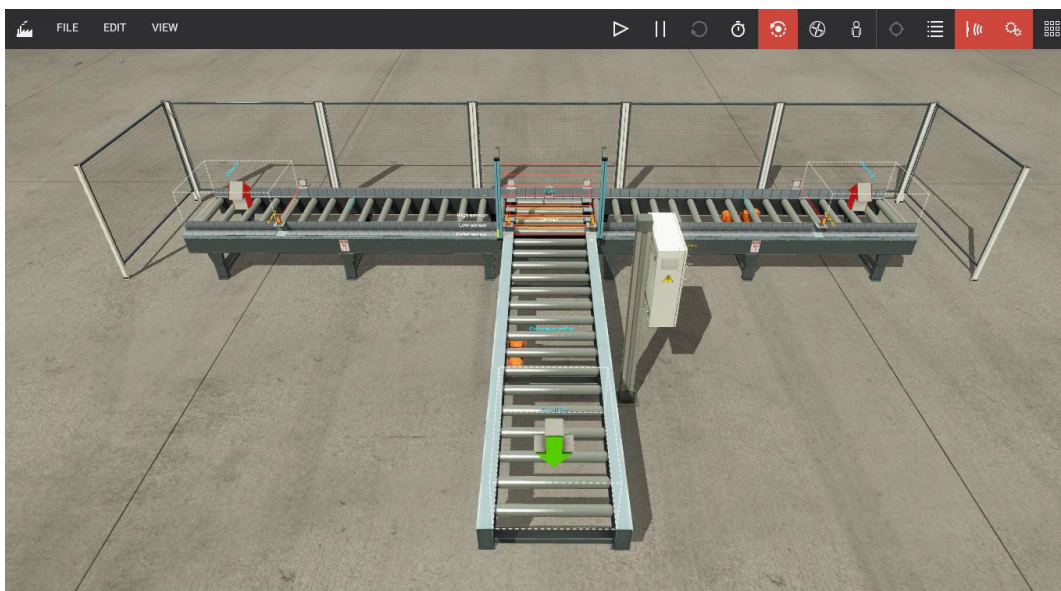


Рис.1. Сцена «Sorting by Height»

2.3. Вивчити принцип роботи технологічного обладнання, поданого у сцені, визначити місцезнаходження та типи приводів і датчиків, їх позначення та змінні, пов'язані з ними. Результати аналізу занести у таблиці (див. табл.1 і 2).

Таблиця 1 Приводи і виконавчі механізми

№ п/п	Призначення	Позначення на сцені	Позначення змінної у програмі	Тип змінної	Стан змінної при запуску
1	Центральний роликовий конвеєр (Переміщує палети)	Conveyor entry	oConveyorEntry	BOOL	True
2	Правий роликовий конвеєр (Переміщує палети)	Conveyor right	oConveyorRight	BOOL	True
3	Лівий роликовий конвеєр (Переміщує палети)	Conveyor left	oConveyorLeft	BOOL	True
4	Транспортер ланцюгового типу (Направляє палету на потрібний конвеєр)	Transf. right Transf. left	oTransfRight oTransfLeft	BOOL BOOL	False False

Таблиця 2 Датчики

№ п/п	Призначення	Позначення на сцені	Позначення змінної у програмі	Тип змінної	Стан змінної при запуску
1	Давач для вимірювання висоти коробок	High sensor Low sensor	iHigh_sensor iLow_sensor	BOOL BOOL	False False
2	Давач виходу палети з ланцюгового транспортера	At right entry At left entry	iAtRightEntry iAtLeftEntry	BOOL BOOL	False False
3	Давач виходу палети з конвеєра	At right exit At left exit	iAtRightExit iAtLeftExit	BOOL BOOL	False False

2.4. Дати словесний опис принципу роботи автоматизованого обладнання.  
Приклад:

При переміщенні коробок на палетах по центральному конвеєрі, здійснюється вимірювання висоти кожної коробки. Коли палета потрапляє на транспортер ланцюгового типу, він направляє її в потрібну сторону (направо або наліво) згідно із спрацюванням датчиків висоти.

Конструктивно транспортер складається із двох приводів – ланцюгового та роликового, що дозволяє йому переміщати об'єкти у чотирьох напрямках. У стані по замовчуванню роликовий привід знаходиться вище ланцюгового (рис.2) , тобто об'єкти рухаються або вперед або назад.

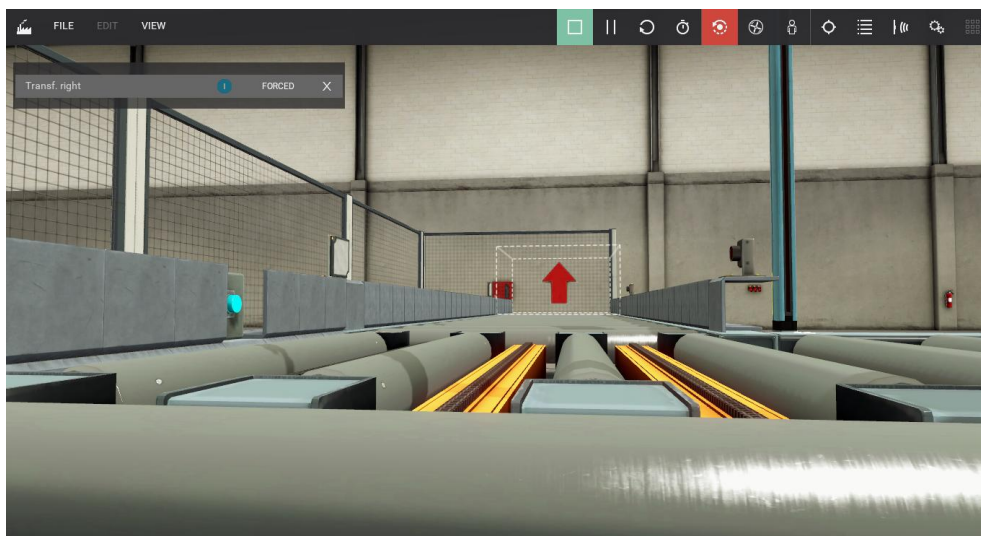


Рис.2. Ланцюговий транспортер у стані по замовчуванню

Коли є потреба у переміщенні об'єктів вправо або вліво , ланцюговий привід піднімається (Рис.3).

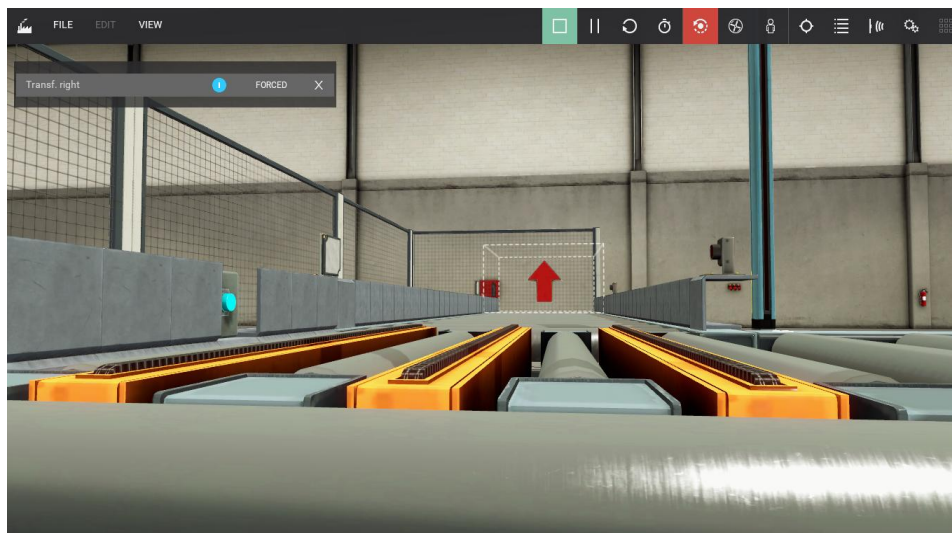


Рис.3. Ланцюговий транспортер з піднятим приводом

Коли палета потрапляє на один з конвеєрів, датчикі (At right entry або At left entry) вказують транспортеру, що на ньому немає палети. Тоді ланцюговий привід опускається, для того щоб на нього могла потрапити нова палета.

2.5. На основі словесного опису розробити блок-схему алгоритму роботи обладнання. Приклад блок-схеми алгоритму подано на рис. 4

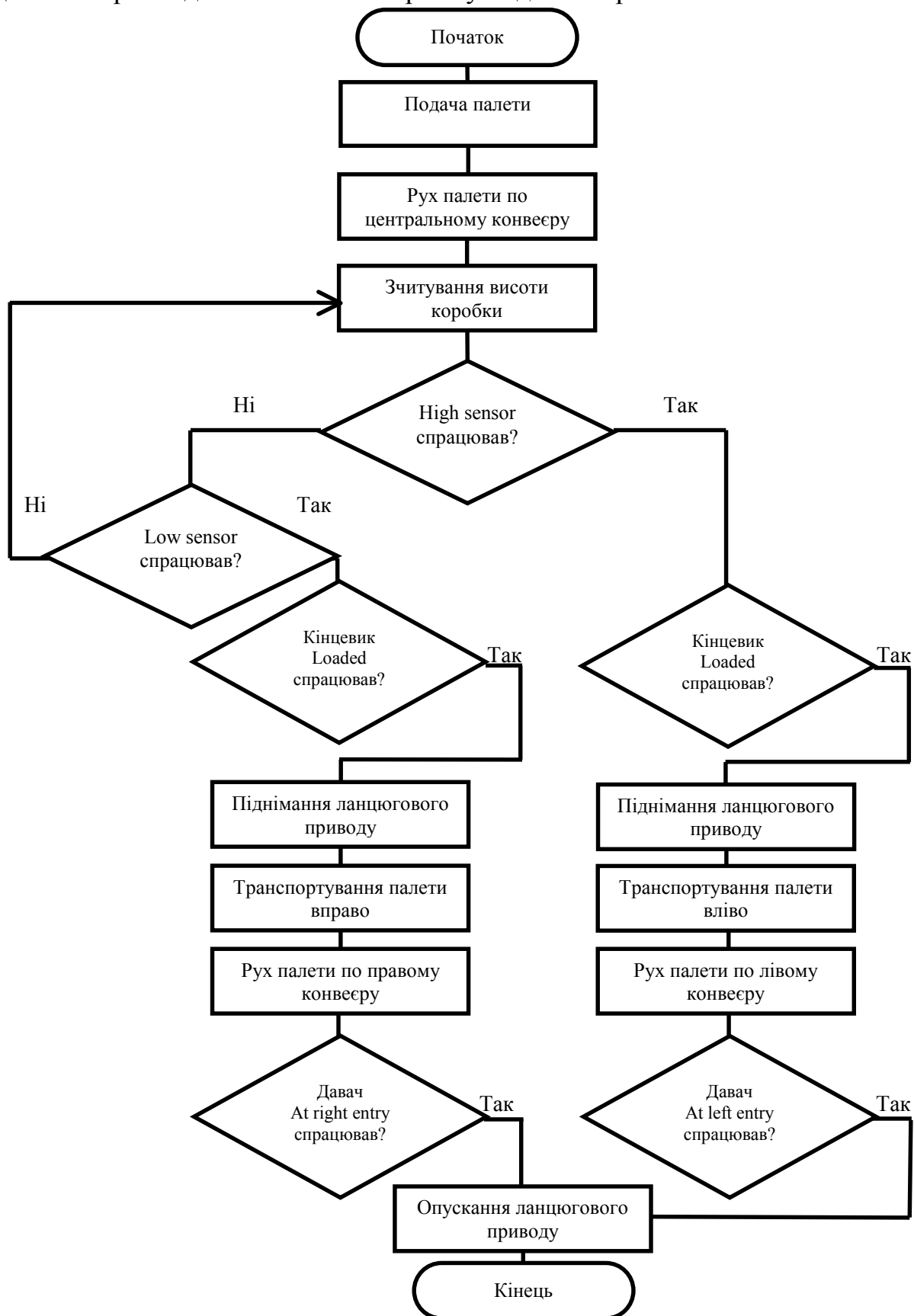


Рис.4. Блок-схема алгоритму роботи обладнання

## Друге заняття

2.6. Після виконня пп. 2.1-2.5 слід запустити CODESYS й створити новий проект (рис.5).

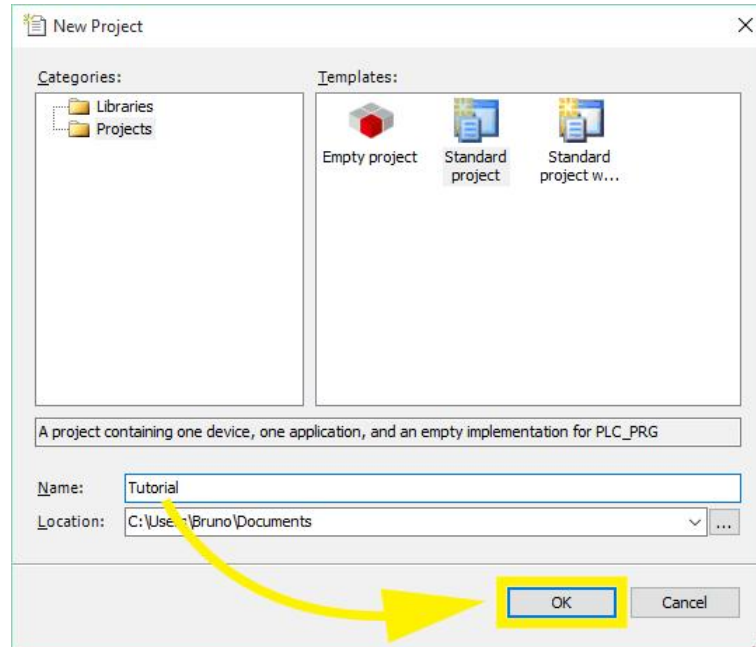


Рис.5. Вікно створення нового проекту

Слід вибирати ярлик **Standard Project** зі списку шаблонів та вказати назву проекту (наприклад **Tutorial**), після чого натиснути **OK**.

2.7. У вікні **Standard Project** (рис.6) потрібно обрати пристрій керування (**Device**) **CODESYS Control Win V3 (3S - Smart Software Solutions GmbH)** та у вікні **PLC\_PRG** мову програмування **Structured Text (ST)**. Натиснути **OK**.

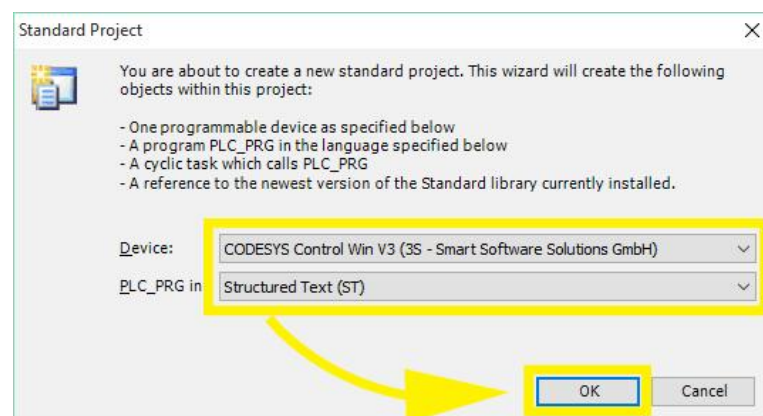


Рис.6. Вікно Standard Project



2.8. Натиснути **ПКМ** на **Application** (рис.7) і вибрати пункт **Add Object > Global variable List...**В списку імен ввести **FIO** та натиснути кнопку **Add**.

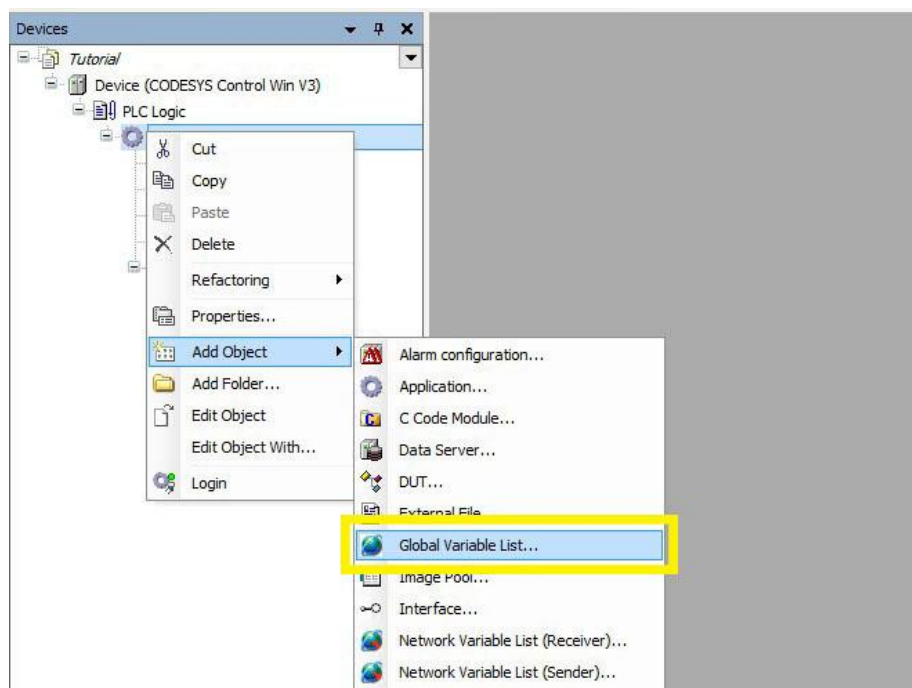
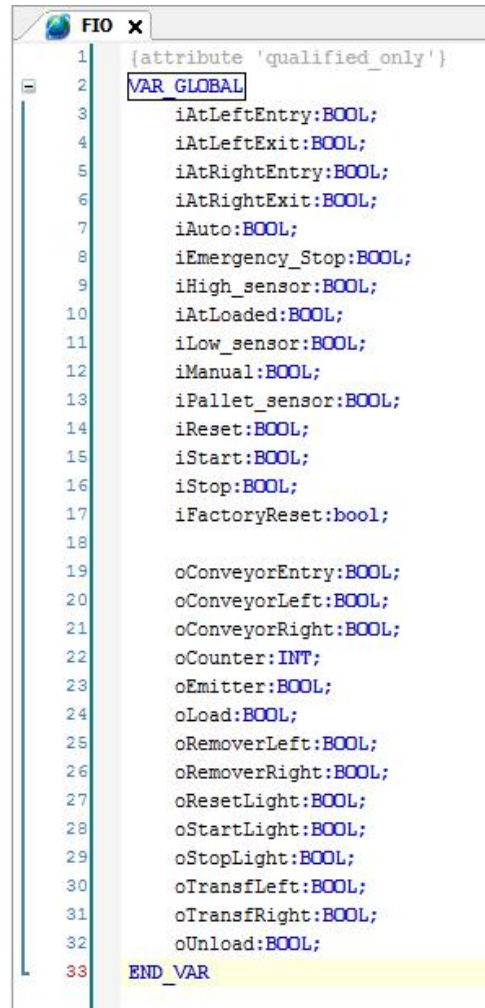


Рис.7. Випадаюче меню пункту Application

2.9. У результаті дій п.2.8 автоматично повинно відкритись вікно FIO. Якщо цього не відбулось то це можна зробити за допомогою **ЛКМ x2** по пункту FIO.

Далі потрібно прописати глобальні змінні. Ці змінні будуть використовувати Factory I/O та CODESYS для обміну даними між собою через OPC Data Access (вони є точками входів / виходів). Для прикладу, що приводиться, вікно FIO буде мати вигляд (рис.8.):



```
1 {attribute 'qualified_only'}
2 VAR GLOBAL
3   iAtLeftEntry:BOOL;
4   iAtLeftExit:BOOL;
5   iAtRightEntry:BOOL;
6   iAtRightExit:BOOL;
7   iAuto:BOOL;
8   iEmergency_Stop:BOOL;
9   iHigh_sensor:BOOL;
10  iAtLoaded:BOOL;
11  iLow_sensor:BOOL;
12  iManual:BOOL;
13  iPallet_sensor:BOOL;
14  iReset:BOOL;
15  iStart:BOOL;
16  iStop:BOOL;
17  iFactoryReset:bool;
18
19  oConveyorEntry:BOOL;
20  oConveyorLeft:BOOL;
21  oConveyorRight:BOOL;
22  oCounter:INT;
23  oEmitter:BOOL;
24  oLoad:BOOL;
25  oRemoverLeft:BOOL;
26  oRemoverRight:BOOL;
27  oResetLight:BOOL;
28  oStartLight:BOOL;
29  oStopLight:BOOL;
30  oTransfLeft:BOOL;
31  oTransfRight:BOOL;
32  oUnload:BOOL;
33 END_VAR
```

Рис.8. Вікно FIO з прописаними глобальними змінними

2.10. Наступним кроком буде написання програми на основі складеного алгоритму роботи для заданого варіанту обладнання. Приклад коду програми для сцени з прикладу («Sorting by Height») можна переглянути у **Додатку А**.

2.11. Після того як програма буде написана , потрібно запустити **CODESYS Control Win V3** (рис.9), відповідний ярлик програми можна знайти на робочому столі.

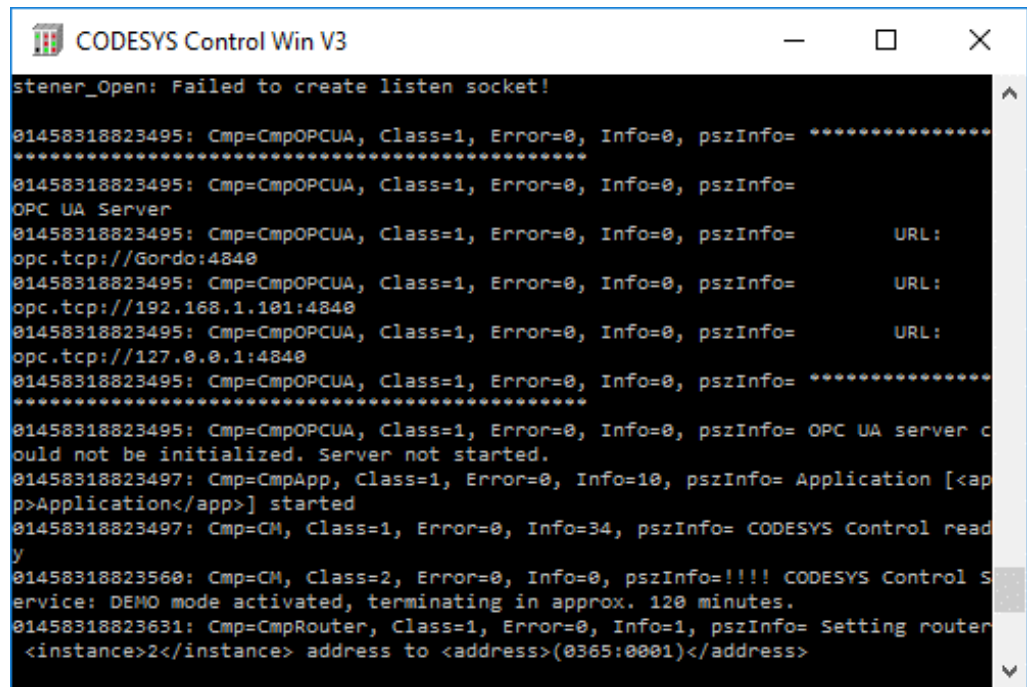


Рис.9. Вікно CODESYS Control Win V3

2.12. Далі потрібно повернутись до вікна CODESYS та перевірити програму на помилки. На панелі інструментів слід вибрати на **Build > Build (F11)**. Якщо виникли помилки, то потрібно допрацювати програму та повторити компіляцію.

2.13. За відсутності помилок слід натиснути на **Online > Login (Alt+F8)**. У впливаючому вікні потрібно підтвердити свої дії, натиснувши **Yes**.

2.14. Далі треба натиснути на **Debug > Start (F5)**. На цьому налаштування коду програми в CodeSys завершено.

2.15. Повернутись до «Factory I/O» та до відповідної сцени («Sorting by Height», рис.1).

2.16. Наступним кроком буде підключення віртуального контролера. Для цього слід натиснути **File** на панелі інструментів та вибирати **Drivers** у впливаючому вікні.

2.17. Обрати **OPC Client Data Access** зі списку драйверів (**Driver list**), як показано на рис.10.

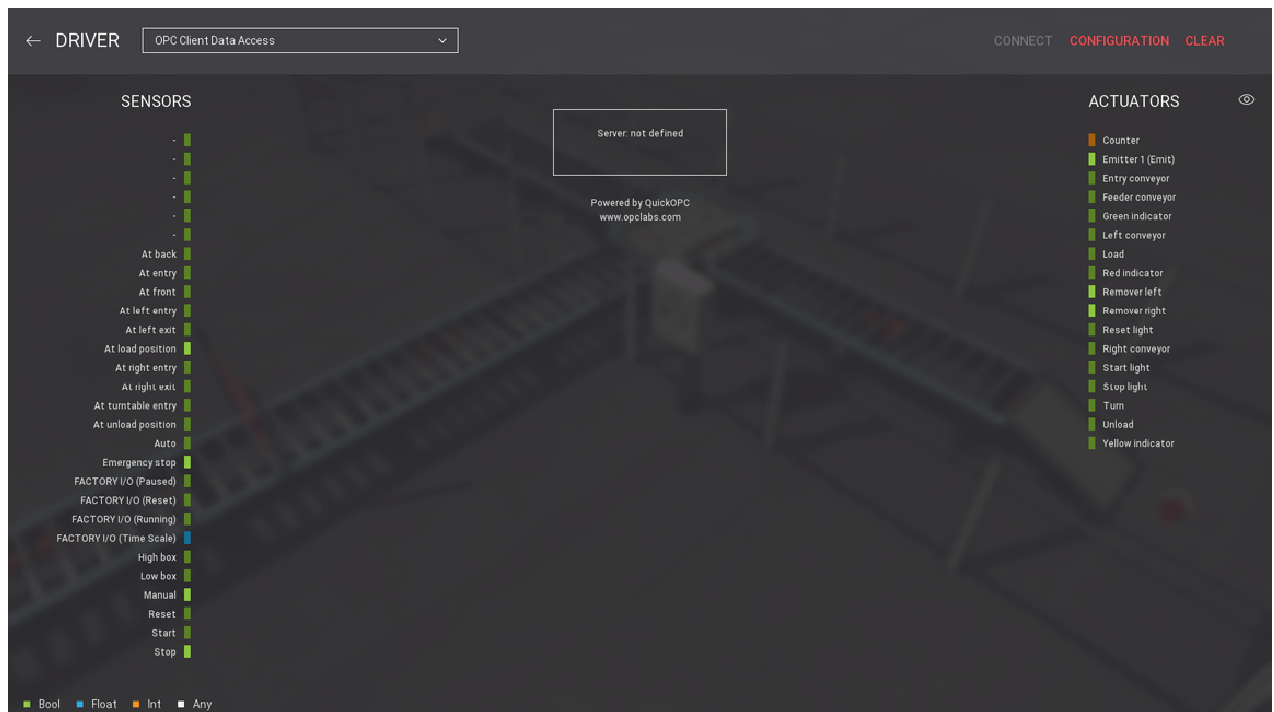


Рис.10. Вікно драйвера

2.18. Натиснути на **CONFIGURATION** та обрати **CoDeSys.OPC.DA** зі списку **OPC** серверів (**OPC Server list**). Після цього натиснути на **BROWSE ITEMS** (рис.11).

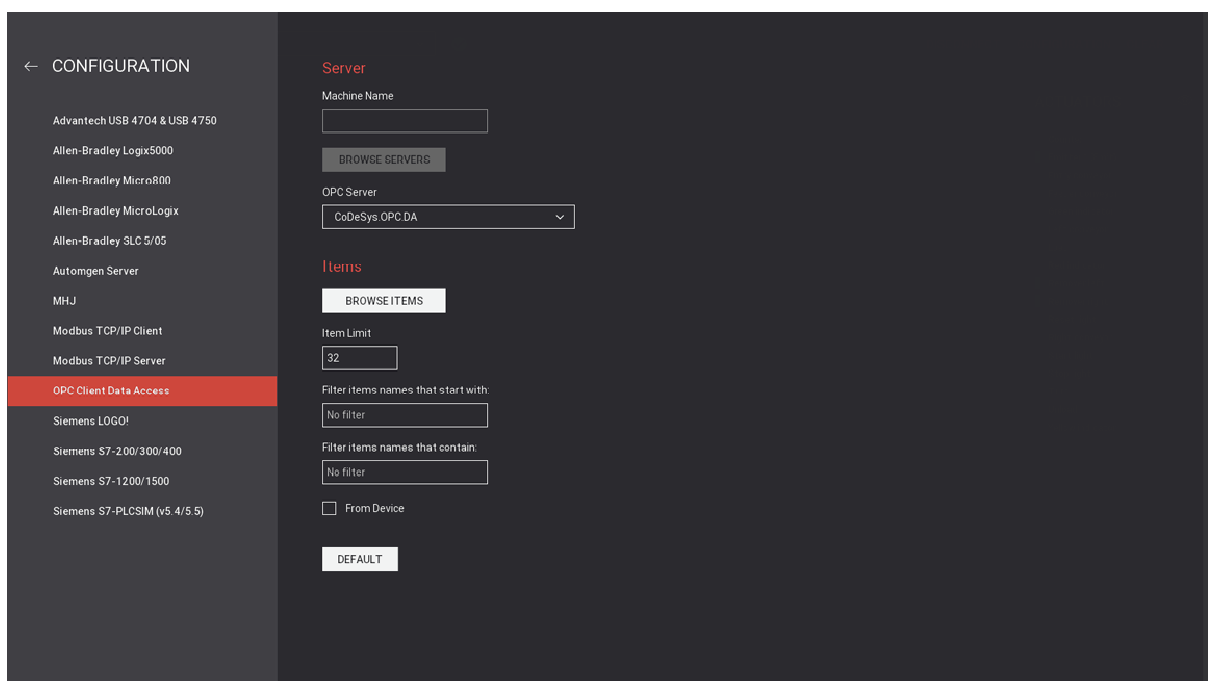


Рис.11. Вікно CONFIGURATION

2.19. Далі потрібно поєднати теги датчиків та виконавчих пристроїв (рис.12) із змінними, які прописувались у програмі.



Рис.12. Зображення з налаштуваннями віртуального контролера

2.20. Після цього потрібно повернутись назад до відповідної сцени та запустити обладнання у роботу (клавіша **F5**). Продемонструвати роботу викладачу, що проводить заняття. У випадку некоректної роботи програми виявити помилки, відкоригувати їх та повторити дії, починаючи з п.2.11.

### 3. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Розробити програму керування для готової сцени згідно варіанту. Сцени є базовими і знаходяться у бібліотеці програми. Варіанти завдань (сцен) можна переглянути у таблиці 3 наведеній нижче. Налагодити сцену на коректну роботу за допомогою програмного коду на мові ST користуючись рекомендаціями розділу 2 даних методичних вказівок.

Таблиця 3 Варіанти завдань

Варіант	Завдання (Сцена)
1	Sorting Station (Сортувальна станція)
2	Palletizer (Станція палетизатор)
3	Pick & Place (Basic) (Станція для переміщення)
4	Separating Station (Розділювальна станція)
5	Sorting by Weight (Сортування по вазі)
6	Assembler (Станція збирання)

#### 4. ЗМІСТ ЗВІТУ

- 1) На титульній сторінці креслярським шрифтом вказати назву закладу, кафедри, назву і номер роботи, прізвище, ініціали, номер групи виконавця, прізвище та ініціали викладача, який керував роботою, рік виконання роботи.
- 2) Вказати тему та мету роботи.
- 3) Подати таблиці з описом виконавчих механізмів і датчиків.
- 4) Дати словесний опис порядку роботи обладнання.
- 5) Подати блок-схему алгоритму роботи.
- 6) Подати текст програми, доповнивши його коментарями.
- 7) Написати висновок до лабораторної роботи.

#### 5. ПЕРЕЛІК ПОСИЛАНЬ

- 1) Методичні вказівки до лабораторної роботи № 21 на тему «Ознайомлення з основами роботи у середовищі програмного забезпечення «Factory I/O» та запуск готового проекту» з курсу «Проектування систем автоматизації» / Шкодзінський О.К., Пісьціо В.П., Сікора Д.А., Герасимів Ю.О. - Тернопіль: ТНТУ, 2018 - 20 с.
- 2) Методичні вказівки до лабораторної роботи № 22 на тему «Модифікація та відлагодження проекту у середовищі програмного забезпечення «Factory I/O»-COSESYS» з курсу «Проектування систем автоматизації» / Шкодзінський О.К., Пісьціо В.П., Сікора Д.А., Герасимів Ю.О. - Тернопіль: ТНТУ, 2018 - 18 с.
- 3) IEC 61131-3:2013 Programmable controllers — Part 3: Programming languages.

- 4) Веб-сторінка фірми «Smart Software Solutions Gmb» виробника середовища CODESYS. Режим доступу: <http://www.3s-software.com/>
- 5) Веб-сторінка ПК "Пролог", підтримка середовища CODESYS російською мовою. Режим доступу: <http://www.codesys.ru/>
- 6) About FACTORY I/O [Електронний ресурс] / NEXT-GEN PLC TRAINING 3D FACTORY SIMULATION. Режим доступу: <https://factoryio.com/docs/>

## Зміст

Тема.....	3
Мета .....	3
1.Теоретичні відомості.....	3
2. Хід роботи .....	3
3. Завдання.....	12
4. Зміст звіту.....	13
5. Перелік посилань.....	13
Зміст .....	15
Додаток А .....	16



## Текст програми для сцени «Sorting by Height»

PROGRAM PLC PRG

VAR

```

    State:INT:=0;
    atState:INT:=1;
    H,L :BOOL;
    stoppressed:BOOL:=FALSE;

```

END VAR

- Опис змінних

IF FIO.iReset THEN

```

    FIO.oCounter:=0;
    FIO.oCounterL:=0;
    FIO.oCounterR:=0;

```

END\_IF

- При натисканні на кнопку Reset всі лічильники будуть скидатись

IF FIO.iFactoryReset THEN

```

    state:=0;
    atstate:=1;

```

END\_IF

- Описування дій які будуть відбуватись при натисканні кнопки «Reset the Simulation»

IF NOT(fio.iEmergency\_Stop) THEN

```

    IF State<>0 THEN
        atstate:=state;
        state:=0;
    END_IF;

```

END\_IF

- Описування дій які будуть відбуватись при натисканні на кнопку екстреної зупинки

IF NOT(fio.iStop) THEN stoppressed:=TRUE;

END\_IF;

- Якщо кнопка Стоп НЕ натиснена то stoppressed є активним

CASE state OF

0:

```

    fio.oUnload:=FALSE;
    fio.oConveyorLeft:=FALSE;
    fio.oConveyorRight:=FALSE;
    fio.oConveyorEntry:=FALSE;
    fio.oEmitter:=TRUE;
    fio.oLoad:=FALSE;
    fio.oRemoverLeft:=TRUE;
    fio.oRemoverRight:=TRUE;
    fio.oResetLight:=FALSE;
    fio.oStartLight:=FALSE;
    fio.oStopLight:=TRUE;
    fio.oTransfLeft:=FALSE;
    fio.oTransfRight:=FALSE;
    fio.oUnload:=FALSE;
    IF fio.iStart THEN
        stoppressed:=FALSE;
        IF atstate<>0
            THEN STATE:=atstate;
            ELSE state:=1;

```

- Опис нульового стану, коли конвеєри виключені

```
END_IF; END_IF;
```

```
1:
  fio.oConveyorLeft:=TRUE;
  fio.oConveyorRight:=TRUE;
  fio.oConveyorEntry:=TRUE;
  fio.oLoad:=TRUE;
  fio.oResetLight:=FALSE;
  fio.oStartLight:=TRUE;
  fio.oStopLight:=FALSE;
  fio.oTransfLeft:=FALSE;
  fio.oTransfRight:=FALSE;
  IF FIO.iPallet_sensor THEN STATE:=2;
  END_IF;
  H:=FALSE;
  L:=FALSE;
```

- Увімкнення конвеєрів

```
2:
  fio.oConveyorLeft:=TRUE;
  fio.oConveyorRight:=TRUE;
  fio.oConveyorEntry:=TRUE;
  fio.oLoad:=TRUE;
  fio.oTransfLeft:=FALSE;
  fio.oTransfRight:=FALSE;
  IF FIO.iHigh_sensor THEN H:=TRUE;
  END_IF

  IF FIO.iLow_sensor THEN L:=TRUE;
  END_IF

  IF NOT(FIO.iPallet_sensor) THEN STATE:=3;
  END_IF;
```

- Опис датчиків висоти та датчика поступлення палет

```
3:
  fio.oConveyorLeft:=TRUE;
  fio.oConveyorRight:=TRUE;
  fio.oConveyorEntry:=FALSE;
  fio.oLoad:=TRUE;
  IF fio.iAtLoaded THEN
    fio.oCounter:=fio.oCounter+1;
    IF H THEN state:=4;
    ELSE state:=7;
    END_IF;
  END_IF
```

- Зупинка центрального конвеєра, завантаження палети на транспортер, опис кінцевика (Loaded), вибір напрямку транспортування (лівий чи правий конвеєр)

```
4:
  fio.oTransfLeft:=TRUE;
  fio.oConveyorLeft:=TRUE;
  fio.oLoad:=FALSE;
  IF NOT(fio.iAtLeftEntry) THEN
    fio.oCounterL:=fio.oCounterL+1;
    state:=5;
  END_IF
```

-Рух палети по лівому конвеєру, збільшення лічильника на одиницю

```
5:
  fio.oConveyorLeft:=TRUE;
  fio.oTransfLeft:=TRUE;
  IF fio.iAtLeftEntry THEN state:=6;
  END_IF
```

- Опис давача At Left Entry

```
6:  fio.oTransfLeft:=FALSE;
    state:=10;
```

- Опускання ланцюгового приводу транспортера

```
7:
  fio.oTransfRight:=TRUE;
  fio.oConveyorRight:=TRUE;
  fio.oLoad:=FALSE;
  IF NOT(fio.iAtRightEntry) THEN
  fio.oCounterR:=fio.oCounterR+1;
  state:=8;
  END_IF
```

- Рух палети по правому конвеєру, збільшення лічильника на одиницю

```
8:
  fio.oTransfRight:=TRUE;
  fio.oConveyorRight:=TRUE;
  IF fio.iAtRightEntry THEN state:=9;
  END_IF
```

- Опис давача At Right Entry

```
9:
  fio.oTransfRight:=FALSE;
  state:=10;
```

- Опускання ланцюгового приводу транспортера

```
10:
  IF stoppressed
  THEN state:=0;
  ELSE state:=1;
  END_IF;
  else state:=0;
END_CASE
```

- Якщо була натиснута кнопка СТОП, то сцена переходить у нульовий стан.

- Якщо нічого не натискалось то сцена переходить у перший стан